REMARKS

Favorable reconsideration of this application as presently amended and in light of the following discussion is respectfully requested.

Claims 10, 12-19, 36, 38-45 and 52-61 are pending in the present application; Claims 11 and 37 having been cancelled, Claims 10 and 36 having been amended, and Claims 52-61 having been added by way of the present amendment.

Before addressing the merits of the outstanding Office Action, it is noted that the Form PTO 1449 which was included with the Official Action of December 8, 1997, did not include an indication that the WIPO document number 91/11871 was considered by the Examiner. The next Official Action is respectfully requested to provide a copy of this form PTO 1449 which indicates that the WIPO document has been considered.

Some of the changes made to the specification were made for the same reasons these changes were made in the parent application. Additionally, the drawing change made herein was made for the same reason it was made in the parent application. The other changes made to the specification include well-known Internet e-mail addressing and format techniques and are supported by the various documents cited in the specification.

In the outstanding Office Action, Claims 10-19 and 36-45 stand rejected under 35 U.S.C. § 103 as being unpatentable over Kraslavsky et al in view of Johnston et al. This rejection is respectfully traversed.

Independent Claim 10 has been amended to include the limitations of Claim 11, and Claim 11 has been cancelled. Similarly, independent Claim 36 has been amended to include the limitations of Claim 37 and Claim 37 has been cancelled. Dependent Claims 11 and 37 whose subject matter was incorporated into the corresponding independent claims pertains to the use of an Internet electronic mail message.

Internet electronic mail messages, according to this specification, are connectionless modes of communication. The outstanding Office Action at the third and fourth lines of page 3 acknowledge that Kraslavsky et al do not explicitly teach transmitting the information by a connectionless mode. However, in numbered paragraph 5 of page 3 of the outstanding Office Action, it is stated that Kraslavsky et al teach the transmitting of the information as an Internet electronic mail message. However, this statement that Kraslavsky et al teach the transmitting of an Internet electronic mail message is clearly erroneous as lines 3 and 4 of page 3 of the outstanding Office Action acknowledge that Kraslavsky et al do not teach the transmitting of information by a connectionless mode. If Kraslavsky et al actually did teach the transmission of an Internet electronic mail message as asserted at numbered paragraph 5, then Kraslavsky et al would teach the concept of a connectionless mode but as acknowledged by the Examiner, Kraslavsky et al do not teach the transmission of a connectionless mode. Therefore, Kraslavsky et al clearly do not teach the transmission of an Internet electronic mail message.

What Kraslavsky et al teach at column 11, lines 13-17, is the sharing of resources from one LAN to another which is the sharing of network resources. However, the mere fact that network resources are shared "internet" does not mean that electronic mail messages are sent. Thus, the rejection of dependent Claims 11 and 37 whose subject matter has been respectively incorporated into independent Claims 10 and 36 is clearly erroneous and therefor, independent Claims 10 and 36 and each of the claims depending therefrom should be allowed.

The added dependent claims recite further features of the electronic mail message and the use of the firewall. For example, dependent Claims 52, 53, 55, 56, 57, 58, 60 and 61 pertain to the format of the electronic mail message. A manner of transmitting Internet electronic mail messages includes an identifier followed by the "@" symbol followed by a domain name. Further, Internet electronic mail messages typically include the description of an encoding type of the electronic mail message. These features of Internet electronic mail are utilized by the BSD Unix mail system which may be used to implement the invention, as described on page 18 of the specification. Further, the specification has been amended to include these features of the BSD Unix mail system. These features are included in conventional email systems and are described, for example, at pp. 441-445 of "TCP/IP Illustrated, Volume 1" by W. Richard Stevens, 1994, which are included herewith as an Appendix. The addition of these features to the specification and claims does not constitute new matter as they are utilized by the BSD Unix mail system and the specification describes that the invention can be implemented by the BSD Unix mail system. It is to be noted that the claims, while reciting features of the Unix mail system, are not limited to the use of a Unix mail system but are limited to general implementations of an Internet electronic mail message system.

As the specific features pertaining to an Internet electronic mail message which are recited in the added claims are neither disclosed nor suggested by the prior art of record, the added dependent claims pertaining to this feature are patentable.

Further, added dependent Claims 54 and 59 and the claims depending therefrom recite the use of a firewall through which the Internet electronic mail message is transmitted. The use of a firewall is useful as it may be used to reject direct connections with the destination but may be configured to allow electronic mail messages which are typically harmless to pass therethrough. Thus, the present invention achieves an advantage by utilizing the combination of a firewall and electronic mail message in order to transmit the information, as set forth in Claims 54 and 59 and the claims depending therefrom.

Accordingly, each of the added dependent claims is patentable over the prior art of record.

Consequently, in light of the above discussion and in view of the present amendment, the present application is in condition for formal allowance and an early and favorable action to that effect is requested.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Gregory J. Maier
Attorney of Record
Registration No. 25,599
James J. Kulbaski
Attorney
Registration No. 34,648

Crystal Square Five - Fourth Floor
1755 South Jefferson Davis Highway
Arlington, VA 22202
(703) 413-3000
Fax #: (703) 413-2220
JJK/gms
I:\atty\JJK\52440051.AM1

Appendix
Serial No. 08/738,659

"TCP/IP
Illustrated Volume I"
W. Richard Stevens
1994 Addison-Wesley

28 SMTP: Simple Mail Transfer Protocol

.1 Introduction

Electronic mail (e-mail) is undoubtedly one of the most popular applications. [Caceres et al. 1991] shows that about one-half of all TCP connections are for the *Simple Mail Transfer Protocol*, SMTP. (On a byte count basis, FTP connections carry more data.) [Paxson 1993] found that the average mail message contains around 1500 bytes of data, but some messages contain megabytes of data, because electronic mail is sometimes used to send files.

Figure 28.1 shows an outline of e-mail exchange using TCP/IP.

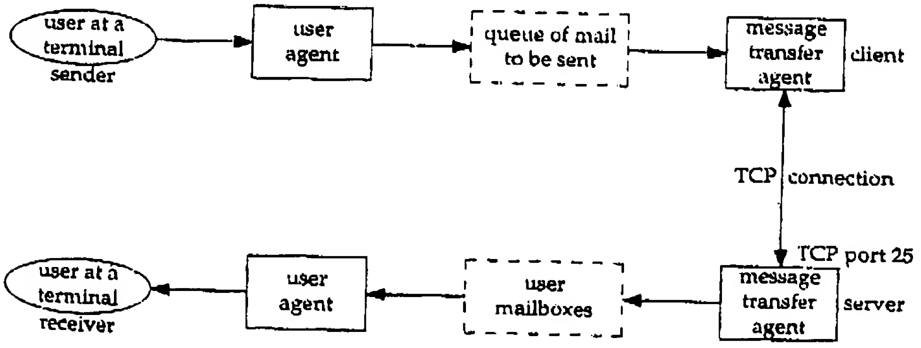


Figure 28.1 Outline of Internet electronic mail.

Users deal with a *user agent*, of which there are a multitude to choose from. Popular user agents for Unix include MH, Berkeley Mail, Elm, and Mush.

The exchange of mail using TCP is performed by a *message transfer agent* (MTA). The most common MTA for Unix systems is Sendmail. Users normally don't deal with the MTA. It is the responsibility of the system administrator to set up the local MTA. Users often have a choice, however, for their user agent.

This chapter examines the exchange of electronic mail between the two MTAs using TCP. We do not look at the operation or design of user agents.

RFC 821 [Postel 1982] specifies the SMTP protocol. This is how two MTAs communicate with each other across a single TCP connection. RFC 822 [Crocker 1982] specifies the format of the electronic mail message that is transmitted using RFC 821 between the two MTAs.

28.2 SMTP Protocol

The communication between the two MTAs uses NVT ASCII. Commands are sent by the client to the server, and the server responds with numeric reply codes and optional human-readable strings. This is similar to what we saw with FTP in the previous chapter.

There are a small number of commands that the client can send to the server: less than a dozen. (By comparison, FTP has more than 40 commands.) Rather than describing each one, we'll start with a simple example to show what happens when we send mail.

Simple Example

We'll send a simple one-line message and watch the SMTP connection. We invoke our user agent with the -v flag, which is passed to the mail transport agent (Sendmail in this case). This MTA displays what is sent and received across the SMTP connection when this flag is specified. Lines beginning with >>> are commands sent by the SMTP client, and lines beginning with a 3-digit reply code are from the SMTP server. Here is the interactive session:

```
sun % mail -v rstevens@noao.edu          invoke our user agent
To: rstevens@noao.edu                    this is output by user agent
Subject: testing                         we're then prompted for a subject
                                         user agent adds one blank line between headers and body
1, 2, 3.                                 this is what we type as the body of the message
.                                        we type a period on a line by itself to say we're done
Sending letter ... rstevens@noao.edu...     verbose output from user agent
                                            following is output by MTA (Sendmail)
Connecting to mailhost via ether...
Trying 140.252.1.54...  connected.
220 noao.edu Sendmail 4.1/SAG-Noao.G89 ready at Mon, 19 Jul 93 12:47:34 MST

>>> HELO sun.tuc.noao.edu.
250 noao.edu Hello sun.tuc.noao.edu., pleased to meet you

>>> MAIL From:<rstevens@sun.tuc.noao.edu>
250 <rstevens@sun.tuc.noao.edu>... Sender ok
```

```
>>> RCPT To:<rstevens@noao.edu>
250 <rstevens@noao.edu>... Recipient ok

>>> DATA
354 Enter mail, end with "." on a line by itself

>>> .
250 Mail accepted

>>> QUIT
221 noao.edu delivering mail

rstevens@noao.edu... Sent
sent.                                  this is output by user agent
```

Only five SMTP commands are used to send the mail: HELO, MAIL, RCPT, DATA, and QUIT.

We type mail to invoke our user agent. We're then prompted for a subject, and after typing that, we type the body of the message. Typing a period on a line by itself completes the message and the user agent passes the mail to the MTA for delivery.

The client does the active open to TCP port 25. When this returns, the client waits for a greeting message (reply code 220) from the server. This server's response must start with the fully qualified domain name of the server's host: noao.edu in this example. (Normally the text that follows the numeric reply code is optional. Here the domain name is required. The text beginning with Sendmail is optional.)

Next the client identifies itself with the HELO command. The argument must be the fully qualified domain name of the client host: sun.tuc.noao.edu.

The MAIL command identifies the originator of the message. The next command, RCPT, identifies the recipient. More than one RCPT command can be issued if there are multiple recipients.

The contents of the mail message are sent by the client using the DATA command. The end of the message is specified by the client sending a line containing just a period. The final command, QUIT, terminates the mail exchange.

Figure 28.2 is a time line of the SMTP connection between the sender SMTP (the client) and the receiver SMTP (the server). We have removed the connection establishment and termination, and the window size advertisements.

The amount of data we typed to our user agent was a one-line message ("1, 2, 3."), yet 393 bytes of data are sent in segment 12. The following 12 lines comprise the 393 bytes that are sent by the client:

```
Received: by sun.tuc.noao.edu. (4.1/SMI-4.1)
        id AA00502; Mon, 19 Jul 93 12:47:32 MST
Message-Id: <9307191947.AA00502@sun.tuc.noao.edu.>
From: rstevens@sun.tuc.noao.edu (Richard Stevens)
Date: Mon, 19 Jul 1993 12:47:31 -0700
Reply-To: rstevens@noao.edu
X-Phone: +1 602 676 1676
X-Mailer: Mail User's Shell (7.2.5 10/14/92)
To: rstevens@noao.edu
Subject: testing

1, 2, 3.
```

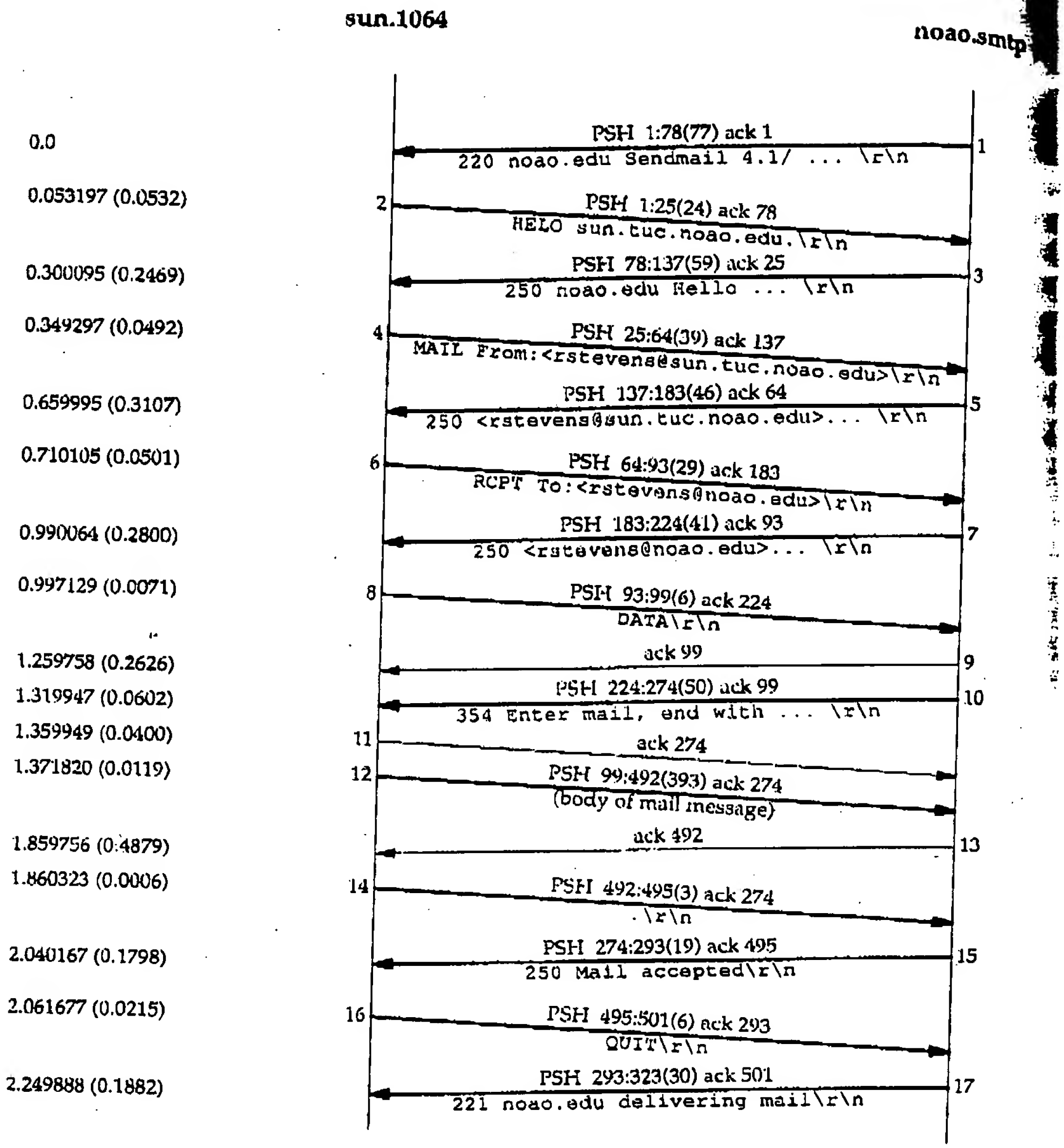
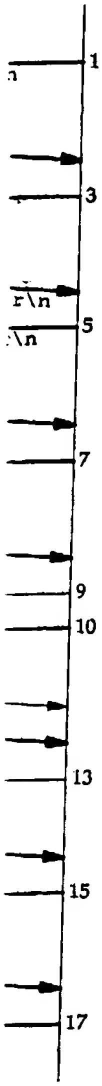


Figure 28.2 Basic SMTP mail delivery.

The first three lines, `Received:` and `Message-Id:`, are added by the MTA, and the next nine are generated by the user agent.

noao.smtp



A, and the

SMTP Commands

The minimal SMTP implementation supports eight commands. We saw five of them in the previous example: HELO, MAIL, RCPT, DATA, and QUIT.

The RSET command aborts the current mail transaction and causes both ends to reset. Any stored information about sender, recipients, or mail data is discarded.

The VRFY command lets the client ask the sender to verify a recipient address, without sending mail to the recipient. It's often used by a system administrator, by hand, for debugging mail delivery problems. We'll show an example of this in the next section.

The NOOP command does nothing besides force the server to respond with an OK reply code (200).

There are additional, optional commands. EXPN expands a mailing list, and is often used by the system administrator, similar to VRFY. Indeed, most versions of Sendmail handle the two identically.

> Version 8 of Sendmail in 4.4BSD no longer handles the two identically. VRFY does not expand aliases and doesn't follow `.forward` files.

The TURN command lets the client and server switch roles, to send mail in the reverse direction, without having to take down the TCP connection and create a new one. (Sendmail does not support this command.) There are three other commands (SEND, SOML, and SAML), which are rarely implemented, that replace the MAIL command. These three allow combinations of the mail being delivered directly to the user's terminal (if logged in), or sent to the recipient's mailbox.

Envelopes, Headers, and Body

Electronic mail is composed of three pieces.

1. The *envelope* is used by the MTAs for delivery. In our example the envelope was specified by the two SMTP commands:

   ```
   MAIL From:<rstevens@sun.tuc.noao.edu>
   RCPT To:<rstevens@noao.edu>
   ```

   RFC 821 specifies the contents and interpretation of the envelope, and the protocol used to exchange mail across a TCP connection.

2. *Headers* are used by the user agents. We saw nine header fields in our example: `Received`, `Message-Id`, `From`, `Date`, `Reply-To`, `X-Phone`, `X-Mailer`, `To`, and `Subject`. Each header field contains a name, followed by a colon, followed by the field value. RFC 822 specifies the format and interpretation of the header fields. (Headers beginning with an `X-` are user-defined fields. The others are defined by RFC 822.) Long header fields, such as `Received` in the example, are folded onto multiple lines, with the additional lines starting with white space.

3. The *body* is the content of the message from the sending user to the receiving user. RFC 822 specifies the body as lines of NVT ASCII text. When transferred